REMARKS/ARGUMENTS

1. Claim Amendments

The Applicant has amended Claims 1-19. Accordingly, Claims 1-19 are pending in the application. Favorable reconsideration of the application is respectfully requested in view of the foregoing amendments and the following remarks.

2. Claim Rejections – 35 U.S.C. §101

The Examiner rejected claims 13, 15 and 19 on the asserted basis that those claims are not directed to statutory subject matter. Claims 13, 15 and 19 have been amended to direct them to statutory subject matter.

3. Claim Rejections – 35 U.S.C. § 103 (a)

The Examiner rejected claims 1-19 under 35 U.S.C. § 103(a) as being unpatentable over Cyran et al (EP 0943990 A), hereafter Cyran, in view of "Automatic Inference of Models for Statistical Code Compression" (Fraser, 1999) Applicant respectfully traverses the rejection, although the Applicant has amended claims 1-19 to better define the intended scope of the claimed invention.

These two references, alone and in combination, do not disclose, nor are they suggestive of, each of the claimed elements in independent claims 1, 12, 14 and 16-19. At a minimum, these references do not disclose, from claim 1: generating, by an encoding stage, a compressed intermediate representation; from claim 12: encoding the transformed code and the compiler information using the extracted state information and statistical information and resulting in a compressed intermediate representation of the input code...; from claim 14: receiving a compressed intermediate representation of an input code, the compressed intermediate representation including encoded transformed code generated and at least partially optimized by a compiler and encoded compiler information indicative of further information generated by the compiler about the transformed code...; from claim 16: an encoding module operable to generate a compressed intermediate representation of an input code, the encoding module further comprising...; from claim 17: encoding means operable to encode the transformed code

and the compiler information using the extracted state information and statistical information and resulting in a compressed intermediate representation of the input code, the compressed intermediate representation being operable to be decoded and further compiled in a subsequent decoding stage for generating executable program code...;from claim 18: receiving means operable to receive a compressed intermediate representation of an input code, the compressed intermediate representation including encoded transformed code generated and at least partially optimized by a compiler and encoded compiler information generated by the compiler indicative of further information generated by the compiler...; and from claim 19: a compressed intermediate representation of an input code, the compressed intermediate representation including encoded transformed code generated and at least partially optimized by a compiler and encoded compiler information indicative of further information generated by the compiler.

According to the Examiner, Figure 1 of Cyran shows a method of generating executable program code for a data processing system. Applicant notes that Figure 1 of Cyran shows a system that takes as input either source code or executable code, and creates optimization information that is used by a runtime environment to execute the code more efficiently. The code preparation system may also generate executable code from source code but the actual method to generate the executable code is not covered by Cyran.

According to the Examiner, the "code preparation system 12" as seen in Figure 1 and described in page 2, line 33 of Cyran is equivalent to the encoding stage of the present invention. Applicant disagrees. The "code preparation system 12" in Cyran, is, at best, equivalent to the first part of an encoding stage. It does not include any compression operations. This is clear as the "extended class file 14" is not compressed, see Cyran page 3 lines 5-13. The distinction is important: "optimization" changes a file of code with a given syntax (e.g. Java byte code) into an equivalent file of code with the same syntax, but which is smaller in size and/or more efficient to run on a computer, whereas "compression" transforms the code into a binary format which is

also smaller in size but does not preserve the syntax. This format can be decompressed to the original format again.

According to the Examiner, the "extended class file 14" and "input code 11j" as seen in Figure 1 of Cyran is equivalent to the step of generating a compressed intermediate representation of an input code. This is not correct as noted above. The "extended class file 14" is at best equivalent to the "transformed code 302" and "compiler information 303".

According to the Examiner, the step, in the present application, of transforming the input code including performing a selected set of code optimization steps is seen in Cyran at page 3, lines 7-10: ("the present invention is a code preparation system 12 which accepts as input pre-processed code 11, analyzes the results, and then provides a code interpretive runtime environment ... with optimization information, hints and/or directions (collectively referred to as "optimization information")) Applicant agrees that Cyran's code preparation system may perform code optimization steps. However, as noted above, Cyran is absolutely silent with respect to disclosing any operations relating to compression or compressed data.

According to the Examiner, the step in the present application of using the input code in further processing of the intermediate code resulting in transformed code and compiler information about the transformed code is seen in Cyran at page 3, lines 12-13: ("optimization Information" in the form of "additional attributes added to class files 14"). Applicant agrees that this is roughly equivalent. However, as noted above, Cyran is absolutely silent with respect to disclosing any operations relating to compression or compressed data.

The Examiner further states that Cyran shows a decoding stage for generating the executable program code from the compressed intermediate representation at Figure 1, ("code interpretive runtime environment") and as described in page 2 lines 38-39: ("The code interpretive runtime environment is operable to use the instructions to further process the intermediate code on the first data processing platform"). Applicant disagrees as the code interpretive runtime environment does not operate on a compressed representation.

The Examiner states that Cyran also shows that the decoding stage further comprises further compiling (407) the transformed code using the decoded compiler information and resulting in the executable program code (EXE) at page 3, lines 15-17: ("by further processing in accordance with the optimization information provided by the code preparation system 12, the code interpretive runtime environment is able to execute the intermediate code mode efficiently...") Applicant agrees that this is roughly equivalent. However, as noted above, Cyran is absolutely silent with respect to disclosing any operations relating to compression or compressed data.

The Examiner acknowledges that Cyran does not disclose that the encoding stage uses a statistical model that is used to encode the transformed code and the compiler information to form the compressed intermediate representation. As noted above, Cyran is absolutely silent with respect to disclosing any operations relating to compression or compressed data.

The Examiner further acknowledges that although Cyran teaches decoding an intermediate representation, Cyran does not show that the decoding stage comprises decoding the compressed intermediate representation (that is, the intermediate representation encoded with the statistical information). Again, as noted above, Cyran is absolutely silent with respect to disclosing any operations relating to compression or compressed data.

To overcome these deficiencies, The Examiner states that Fraser shows a method of compressing computer programs, and in particular of intermediate representations.

According to the Examiner, Fraser shows that state information (e.g. the last few tokens seen, stack height, data type of the top few stack elements, see page 243 section "IR predictors) and statistical information (a decision tree is generated and a probability distributed to each leaf, see page 243 section "Background: Machine learning of decision trees") is extracted from the transformed code and the compiler information.

Applicant disagrees. Fraser does not disclose the use of compiler information, rather only states that information can be inferred from the intermediate representation,

although the "computed predictors" and "reduced predictors" could be included in the compiler information.

The Examiner also states that Fraser shows that the extracted state information and statistical information are used to encode the transformed code and compiler information, resulting in a compressed intermediate representation (The Examiner refers to page 242, "Motivation": "This papers principal focus is ... the more basic problem of statistical models that reduce entropy, because such models lead directly to a variety of compact encodings"'). Although Fraser is not specific about the statistical information, he makes use of it, for instance in the MTF coder which is based on statistical information.

The Examiner also says that Fraser shows that the compressed intermediate representation is decoded resulting in the transformed code and the compiler information (see at least page 242, "Motivation": "saving even a few percent in size frees up more than enough resources to implement the decompressor").

CONCLUSION

In view of the foregoing remarks, the Applicant believes all of the claims currently pending in the Application to be in a condition for allowance. The Applicant, therefore, respectfully requests that the Examiner withdraw all rejections and issue a Notice of Allowance for all pending claims.

The Applicant requests a telephonic interview if the Examiner has any questions or requires any additional information that would further or expedite the prosecution of the Application.

Respectfully submitted,

By Michael G. Cameron
Registration No. 50,298

Date: November 10, 2009

Ericsson Inc.
6300 Legacy Drive, M/S EVR 1-C-11
Plano, Texas 75024

(972) 583-4145
michael.cameron@ericsson.com